Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1. (Currently amended) A method in a host machine for validating a design for a system which comprises a software element and first and second hardware components, the software element being for execution on the second hardware component and the first and second hardware components being operable to interact with one another, the method comprising the steps of:

simulating operation of the first hardware component in a first software simulation system;

simulating the software element and the second hardware component in a second software simulation system;

receiving a variable synchronization parameter;

running the second software simulation system asynchronously with, and ahead of, the first software simulation system, wherein the second software simulation system advances at most by a number of processor clock cycles set in the variable synchronization parameter before the first software simulation system advances by a clock cycle, the variable synchronization parameter limiting a maximum number of processor clock periods of the second simulation per period of a reference clock of the host machine;

setting up an inter-process communications protocol connection in the first software simulation system, the inter-process communications protocol being configured to connect to the second software simulation system;

controlling the first software simulation system using the second software simulation system that is running ahead of the first software simulation system, a socket allowing for communication between the second software simulation system and the first software simulation system; and

analyzing a result from the first and second software simulation systems and validating the design for the system,

wherein the first software simulation system and the second software simulation system are implemented in separate processing threads within the host machine providing more rapid simulation of software instructions in the second software simulation system than the simulation of instructions in the first software simulation system.

Claims 2-4. (Cancelled)

Claim 5. (Previously presented) A method as claimed in claim 1, further comprising:

connecting the second software simulation system to the inter-process communications protocol connection in the first software simulation system;

connecting a software debugger to the second software simulation system; and

controlling the first software simulation system from the software debugger via the second software simulation system using the inter-process communications protocol.

Claim 6. (Previously presented) A method as claimed in claim 1, further comprising:

connecting a software debugger to the inter-process communications protocol connection; and

controlling the first software simulation system from the software debugger using the inter-process communications protocol.

Claim 7. (Original) A method as claimed in claim 5 or 6, wherein the inter-process communications protocol is TCP/IP and the connection is a TCP/IP socket.

Claim 8. (Original) A method as claimed claim 1, wherein the second hardware component includes a processor.

Claim 9. (Original) A method as claimed in claim 8, wherein the processor is an embedded processor.

Claim 10. (Previously presented) A method as claimed in claim 1, wherein the second hardware component includes processor peripheral devices.

Claim 11. (Original) A method as claimed in claim 10, wherein the peripheral devices are embedded.

Claim 12. (Previously presented) A method as claimed in claim 1, wherein the first software simulation system is implemented using a hardware description language (HDL) simulation environment.

Claim 13. (Previously presented) A method as claimed in claim 1, wherein the second software simulation system is implemented using a C model.

Claim 14. (Original) A method as claimed in claim 1, wherein the first hardware component is a programmable logic device.

Claim 15. (Previously presented) A method in a hardware environment for controlling a simulation of a system using a software debugger, the simulation useful for validating a design of the system, wherein the system comprises a software element and first and second hardware components, the software element being for execution on the second hardware component and the first and second hardware components being operable to interact with one another, the method comprising the steps of:

simulating the first hardware component in a first software simulation in the hardware environment;

simulating the software element and the second hardware component in a second software simulation using a software model embedded within the hardware environment, the first software simulation and the second software simulation being implemented in separate processing threads within the hardware environment;

setting up an inter-process communications protocol connection in the first software simulation, wherein the inter-process communications protocol connects to the second software simulation;

connecting the software debugger to the software model of the second software simulation embedded in the hardware environment;

receiving a variable synchronization parameter;

running the second software simulation asynchronously with, and ahead of, the first software simulation, wherein the second software simulation advances at most by a number of processor clock cycles set in the variable synchronization parameter before the first software simulation advances by a clock cycle, the variable synchronization parameter limiting a maximum number of processor clock periods of the second simulation per period of a reference clock of the hardware environment;

controlling the first software simulation of the first hardware component from the software debugger through the second software simulation using the inter-process communications protocol; and

validating the design of the system using the first and second software simulations.

Claim 16. (Original) A method as claimed in claim 15, further comprising the step of:

connecting the software debugger to inter-process communications protocol connection.

Claim 17. (Cancelled)

Claim 18. (Original) A method as claimed in claim 15, wherein the inter-process communications protocol is TCP/IP and the connection is a TCP/IP socket.

Claim 19. (Original) A method as claimed in claim 15, wherein the step of simulating the second hardware component comprises simulating a processor and one or more peripheral devices with which the one or more processors interact directly.

Claims 20-23. (Cancelled)

Claim 24. (Original) A method as claimed in claim 15, wherein the second hardware component includes embedded processors.

Claim 25. (Original) A method as claimed in claim 15, wherein the second hardware component includes embedded peripheral devices.

Claim 26. (Original) A method as claimed in claim 15, wherein the first simulation is implemented using a hardware description language (HDL) simulation environment.

Claim 27. (Original) A method as claimed in claim 15, wherein the second simulation is implemented using a C model.

Claim 28. (Original) A method as claimed in claim 15, wherein the first hardware component is a programmable logic device.

Claim 29. (Previously presented) A method for providing an I/O interface for a simulation model to allow the simulation of interactive programs in a hardware environment for use in system validation, the method comprising:

simulating a software element in a first software simulation using a software model in a first processing thread in the hardware environment;

simulating an embedded input/output device within the simulation model in a second software simulation to produce an input/output device model in a second processing thread, the first software simulation running ahead of the second software simulation, the first and second software simulations being synchronized using a reference clock parameter that

limits a maximum number of processor clock periods of the first processing thread per clock period of the second processing thread, wherein the reference clock parameter is selectable;

connecting the input/output device model to a terminal emulator using an inter-process communications protocol;

running an interactive program in the terminal emulator to interact with, and transfer information to, the input/output device model;

setting up an inter-process communications protocol connection in the software model, wherein the inter-process communications protocol connects to the input/output device model;

polling the input/output device model for the transferred information using the inter-process communications protocol connection in the software model; and

validating a design of the system.

Claim 30. (Original) A method as claimed in claim 29, the method further comprising:

providing separate processing threads for the embedded input/output device to allow concurrent user inputs and outputs.

Claim 31. (Original) A method as claimed in claim 29, wherein the inter-process communications protocol is TCP/IP.

Claim 32. (Original) A method as claimed in claim 29, wherein the input/output device is a UART device.

Claim 33. (Original) A method as claimed in claim 29, wherein the input/output device is an Ethernet MAC device.

Claim 34. (Previously presented) The method as claimed in claim 1, wherein the host machine comprises a plurality of processors.

Claim 35. (Previously presented) The method as claimed in claim 34, the method further comprising:

setting a first software simulation variable, wherein the setting specifies synchronous or asynchronous simulation between the first software simulation system and the second software simulation system.

Claim 36. (Previously presented) The method as claimed in claim 35, wherein asynchronous simulation uses thread scheduling of the host machine.

Claim 37. (Previously presented) A method as claimed in claim 1, wherein the controlling the first software simulation system using the second software simulation system comprises the second software simulation system sending control commands to the first software simulation system, to which the first software simulation system responds.

Claim 38. (New) A method for simulating a system which comprises a software element and first and second hardware components, the software element being for execution on the second hardware component, and the first and second hardware components being operable to interact with one another, the method comprising:

receiving a variable synchronization parameter;

simulating operation of the first hardware component in a first simulation;

simulating the software element and the second hardware component in a second simulation; and

running the second software simulation asynchronously with, and ahead of, the first software simulation, wherein the second software simulation advances at most by a number of processor clock cycles set in the variable synchronization parameter before the first software simulation advances by a clock cycle,

wherein the first simulation and the second simulation are implemented in separate processing threads.

Claim 39. (New) The method of claim 38, wherein the variable synchronization parameter limits a maximum number of processor clock periods of the second simulation per period of a reference clock of the host machine

Claim 40. (New) The method of claim 38, wherein the first and second simulation run asynchronously.

Claim 41. (New) The method of claim 38, wherein a number of clock cycles of the first simulation and the second simulation are synchronized with a reference clock.

Claim 42. (New) The method of claim 38 further comprising:

performing operations in the first simulation to set up an inter-process communications protocol connection therein;

connecting a software debugger to the communications protocol connection; and

controlling the first simulation from the software debugger using the inter-process communications protocol.

Claim 43. (New) The method of claim 38, wherein the second hardware component includes embedded processors.

Claim 44. (New) A method for controlling a simulation of a system using a software debugger, wherein the system comprises a software element, and first and second hardware components, the software element being for execution on the second hardware component and the first and second hardware components being operable to interact with one another, the method comprising the steps of:

simulating the first hardware component in a first simulation;

simulating the second hardware component in a second simulation;

performing operations in the first simulation to set up an inter-process communications protocol connection; and

controlling the first simulation from the software debugger using the inter-process communications protocol.

Claim 45. (New) The method of claim 44, further comprising the step of:

connecting the software debugger to the inter-process communications protocol connection.

Claim 46. (New) The method of claim 44, wherein the step of simulating the second hardware component comprises simulating a processor and one or more peripheral devices with which the one or more processors interact directly.

Claim 47. (New) The method of claim 44, wherein the first simulation and the second simulation run asynchronously.

Claim 48. (New) The method of claim 44, wherein the first simulation and the second simulation are synchronized with a reference clock.

Claim 49. (New) The method of claim 44, wherein the second hardware component includes embedded processors.